

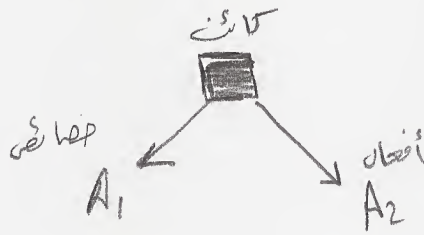
Java 101

Lesson 1

* every thing is an object كل شيء عبارة عن كائن

* every object consists of

Attributes ^{المواصفات} "property" ^{خاصة}
 Actions ^{أفعال}



Example:

Car is an objects

→ Attributes: Model, Color ..
→ Actions: move-turn-fly ..

Color = red, Model = BMW

Lesson 2

Attributes:

الخاصة

* أشياء موجودة مع الكائن (أي، ينتمي إليها)

"المواصفات"

اسم الخاصية

قيمة الخاصية

Name = Value

→ دي كده خاصية

مثال:

age = 45

Example: ^{الخاصة} ^{أشياء موجودة مع الكائن}

name = Ali

age = 20

length = 180

Color = white

Lesson 2

نصفها

الأفعال
Actions

الاشياء التي تقوم بها الكائنات

Ex

Action

object

Car

move
turn
stop

Lesson 3 :-

« الكائنات الكائنات »

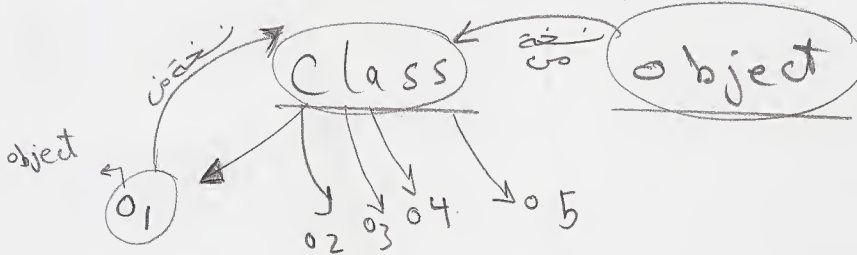
Car

attributes	object name	Actions
A ₁	A ₂	
Color	move	
Name	turn	
Company	stop	
⋮	⋮	

وهذه الكائنات من طريق جدول
أي كائنات تقدر تتناول هذا الشكل
" تصيل كائنات "

Lesson 4 :-

مفهوم يربط الكائنات



* الكائنات نسخة من الكلاس

Ex: Car

وهذه الكائنات

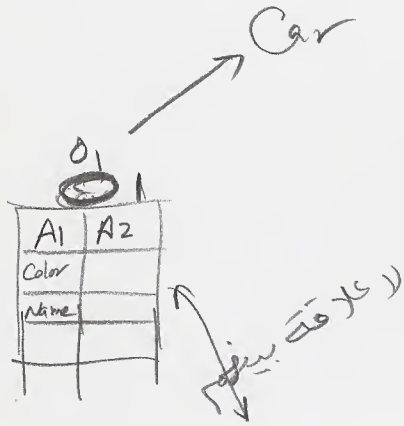
Car Class

A ₁	A ₂
Color	move
Name	turn
Model	stop

This is class
العمل فانهي بدون قيم
قيم فارغة

لا تتلقه كائنات object بوجه يأخذ نسخة من جدول
ويجب بيانات

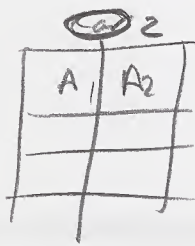
لنعرف Class اسمه Car اقله أعطيني نسخة من Car وسميها 01 .



و كده هتعمل جدول بنافس مع ال Class الى انت عملته
نسخة منه

* بالظبط هتسوخ جدول زي ما هو

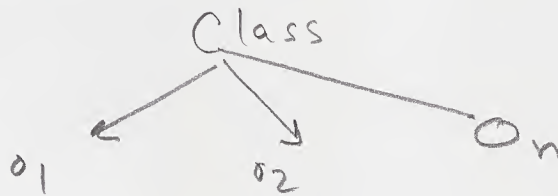
* وسعد كده هيعني القيم



ولو قلنا اعمل نسخة أخرى اسمها 02 :

هتسوخ ناسخ جدول ال Class ويحط تاني ويحط بياناته

* بديري الكلاس لا يوجد شيء اسمه Object لان ال Object هونسخة من ال Class



* الكلاس يُنشأ مرة واحدة وبعد ذلك تُنسخ منه الكائنات

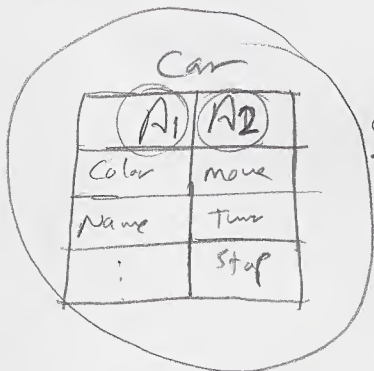
* الكائن ال Object هو عبارة عن نسخة من ال Class

* Lesson 5

مفهوم ال Class

دي كلها كلاس

كلاس : ان تضع الامتياز في تصنيفات "نموذج - صيغ - نبات - انسان"



class car

* لما قبيح تنشأ ال Object للكمبيوتر وهو هيقولك أنا لا أعرف
هكذا الكائن فستظهر انا كائن ال Class للكمبيوتر في اللفظ
عنا هعرف أنواع الكائنات .

عني كده لما اقله انا دة كلاس هيدوع يشوف A1 خصائصه السيارة
و A2 افعال السيارة وكده يكونه اشعرف عليها

* ال كلاس زي خريطة بالنسبة للكمبيوتر "أردليل ارشادي"

لوجيت على الكمبيوتر

Car 01

عني انشاء object جديد اسمه 01 من الكلاس Car

عني هياخذ الجدول بتاع class وبنسحو

بتسويكي القيم بقا عني

Color = red

Model = 1991

Name = BMW

لو جيت على الكمبيوتر

Car 02

02

A1	A2
Black	---
...	...
2002	

02 & 01 →

لا توجد علاقة بينهم

لكن

02 & 02 →

لهم كلاهما مشترك

Lesson 6

البرمجة الشيئية

لغة جافا تصنف مع أنها Object oriented programming
 "oo" "oop"
 عن لغة التعامل به مفهوم البرمجة
 والكائنات

المتغير لما يكون داخل الكلاس
 يسمى خاصية

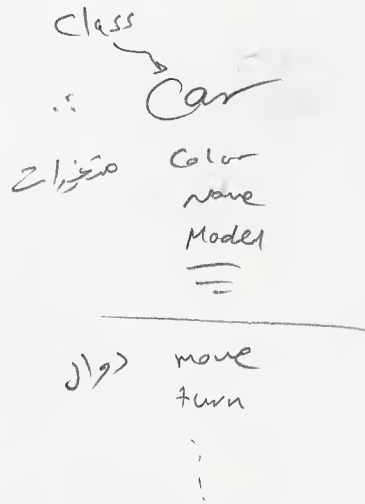
Variables

* خصائص attributes يتم تمثيلها في البرمجة بـ المتغيرات

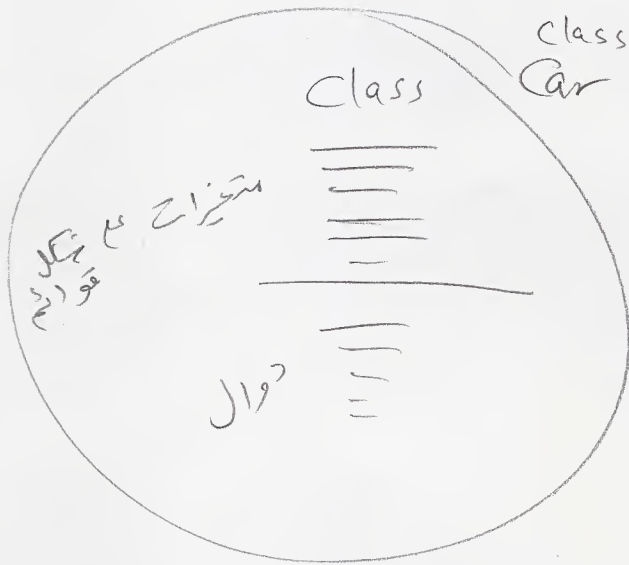
"Methods" Functions

* أفعال actions الروال

البرمجة لما إذا كانت
 تتخذ الكلاس
 تسمى Method



كل كائن سيكون بداخله متغيرات ودوال



Car x ;

Car y ;

← اكد ان object اسمه x من الكلاس Car

كده هينسخ جميع المتغيرات والدوال وهنطوي
 فيها x

Lesson 7

المتطلبات

حوله بلغة جافا
Java
للمبيوتر

البنية Class اعمل جدول النظري

Car	
A ₁	A ₂
Color	—
Name	—

Coding:-

Class car {

int model;

() {

}

Car x;

x.model = 2011

المتطلبات
"Car"
المتطلبات
المتطلبات

Car Z;

المتطلبات المتطلبات

المتطلبات من الكلاس car

المتطلبات المتطلبات "model"

المتطلبات Model=2010 ← "Car"

Lesson 8 : netbeans برنامه نویسی

+ download program ✓

+ public class JavaCourse { ...

class _____

Lesson 9

Ex :-

JavaCourse.java

+ لازم که اسم فایل بنفس اسم ار class خورود داشته

Lesson 10

+ البته اگر داخل الكلاس ← نفس Method

+ الكلاس خريطة لكائنات

Lesson 11 : Comments

+ الكود من كائنات اي هتطلع من الكود

→ Inline Comment : من كود كومت في اكواد

Ex: // welcome to Anan

→ Multi line Comment : الملاحظات كتبت في اكثر من سطر

/* * *

note: /* *

+

↳ @author { name }

*/

Java documentation
دليل البرنامج

Lesson 12

كيف سيخزن الكمبيوتر البيانات
أمنه بالذاكرة ؟

ع/ الكمبيوتر يهيكل كل خانة عنوان
عنوانه يميز

و غالباً ترقيمها بالـ hex & decimal

النظام الست عشري

address

0X 321

0X 322

0X Fc1

الرمز كبري جداً

ذاكرة الكمبيوتر

Content

خانة

10

7

35

هنا تخزن
البيانات

Size
1 byte

> 0X125FC1234DA1 - - - -

In programming

Store :- "تخزين" يخزن في الذاكرة

Fetch :- "جلب" بتجيب من الذاكرة

لعمري اكمل مع مكان معين في الذاكرة
لن نقرر مع العامل معه والابا العنوان

"Store"

0X Fc1 = 35 \Rightarrow روح العنوان ده وعطانيه لتيه 35

العنوان ده كبير جداً
عنه كد 40 جوا المتغيرات

المتغيرات :- أسماء مستقرة العناوين الذاكرة

Ex 1 X = 7 \Rightarrow

الكمبيوتر هيربط X بعنوان
معين بشكل تلقائي

print X

Computer print 0X 322

"Fetch"

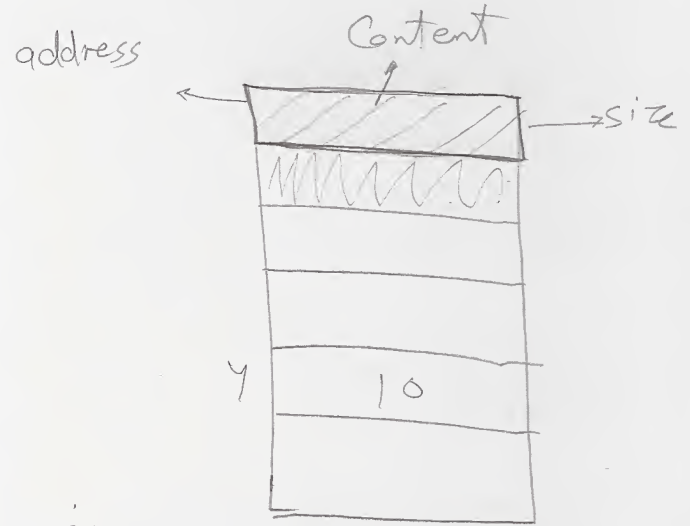
النتج
7

Lesson 13

- * في البرمجة : أنت تتعامل مع الذاكرة عن طريق المتغيرات
- * لا تقدر الوصول للذاكرة بطريقة مباشرة ، لأنك لست تعرف المتغيرات .
- * المتغيرات : وسيلة يستخدم المبرمج للتخزين في الذاكرة والاستفادة منها .

Constraints
بالإمكان وضع قيود على الـ cell المخزن فيها
نص - رقم - عدد باسعة عشرية - ...

Constraints: حدد نوع القيمة ليتمكن من
اد الـ cell



* Data types

Char : حرف character	String	نص
	int	رقم صحيح
	float	
	double	

Ex : int x : الخلية x لا تقبل الا عدد صحيح

Char z : كده عرفت متغير اسمه z
ولازم يكون محتواه حرف "char"

int y = 10 : حجز مكان في الذاكرة اسمه y
ولازم هذه القيمة تكون عدد صحيح

data type	name	=	value
النوع	الاسم		قيمة المتغير

قيمة المتغير في الغالب يجب ان تكون موافقة لنوع البيانات

Lesson 14

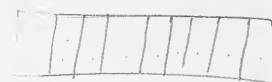
type name = value

الـ size يختلف نوع Type الى نوع

Ex: $\text{int } x = 10$

كل نوع له حجمه في الذاكرة

1 cell = 1 byte = 8 bit



* $\text{int} \rightarrow 4 \text{ cells} \rightarrow 32 \text{ bit}$
 الخلية الواحدة

* float : عدد عشري " عدد بفاصلة عشرية " 4 byte

* double : عدد بفاصلة عشرية أطول 8 byte

Lesson 15

Numerical Data type
 أنواع البيانات العددية

0 → 255

byte = 256

In java :-

byte

-128

127



2 byte

short

عدد صحيح

int

عدد صحيح أدق

long

عدد صحيح أدق

float

عدد بفاصلة عشرية

double

عدد بفاصلة عشرية أدق

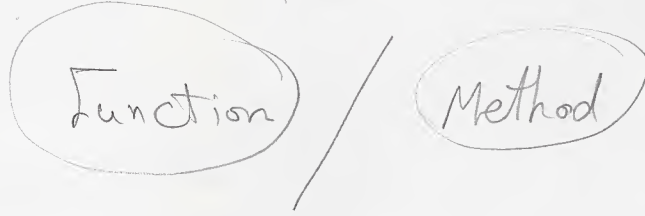
Lesson 16 :-

مفاهيم من المتغيرات

* Declare variable " public int model ;
 اسم المتغير ← نوع البيانات ← يكتبها كده دلوقت وبعدين هتفهمها

Lesson 17 :-

مفهوم الدوال



والدوال هي Function
 بتعني الاسماء فقط بديسموها كده

Class {

Function

المدة اذا جادت بداخل الكلاس
 تسمى Method

}

الكانت له خصائصه فعال

ملف برجي



Actions

تسمى
 Functions

له اسم وبداخله كود برجي



Function
 * المدة عبارة عن Block
 ↓
 Function definition
 تعريف المدة

* البلاك الذي يحتوي على السطر لاسم تنفيذها ولا بد ان اسم استدعائه

لو عايز استدعي الدالة get نكتب

get



get()

استدعاء الدالة Function Call

روح للبلاك رالي اسم الدالة ونفذ الشفرة التي بداخله

Lesson 18

تعريف الدوال

public void Move()

```
{
  // ...
}
```

استدعاء الدوال

Lesson 19

Function definition

التعريف يكتب مرة واحدة

Function call

والا استدعاء يكتب براحتك زي ما قيت طريقة الاستدعاء

استدعاء الدالة

stop()

Lesson 20

معلومات الخلية عن Class

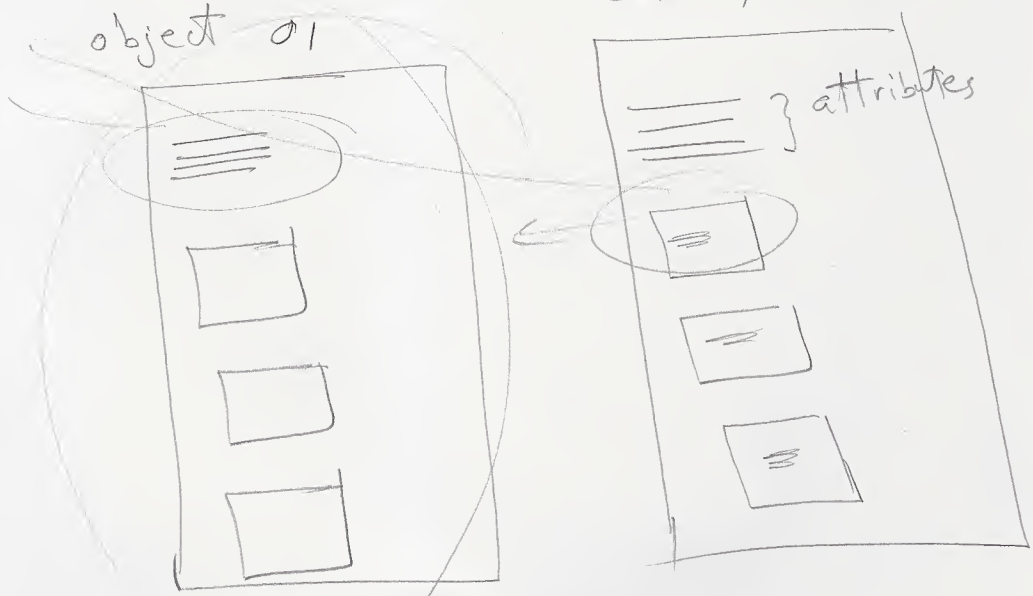
03

02

object 01

Class X

attributes



لا يوجد كود يكتب خارج الدوال "البلوكات"

ملاحظة: لا نضع استدعاء الدالة خارج الكلاس

Lesson 21

مثال بسيط

`System.out.print ("The value of c is" + c + " Kilogram");`

Capital

استخدم علامة زائد لوترتيب المتغيرات
لربط

Lesson 22

إشارة دالة مبسطة للتوضيح

`public void speed ()`

`{ int a, b, c ;`

`int a, b, c; = int a;
int b;
int c;`

`a = 10 ;
b = 20 ;
c = a + b ;`

`System.out.print (c) ;`

`}`

لأن سيم تنفذ هذا الجزء
ألا عند استدعائه
`speed()`

Lesson 23

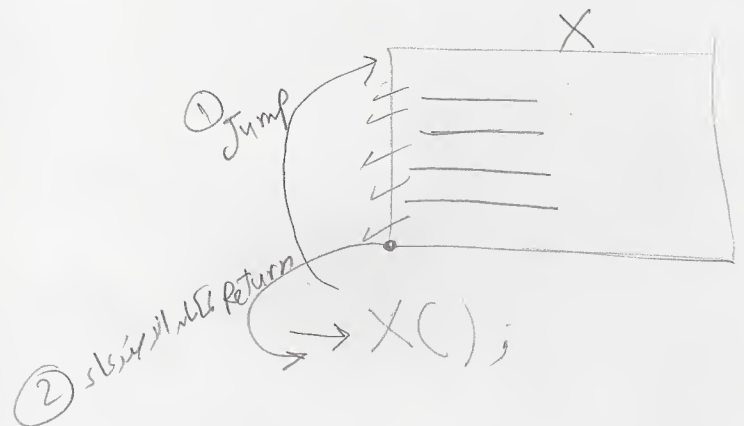
نوع الدالة

`type name ()`

`{`

`}`

لا تكتب الدالة هكذا



```

type name ( )
{
}

```

ex: int nam ()

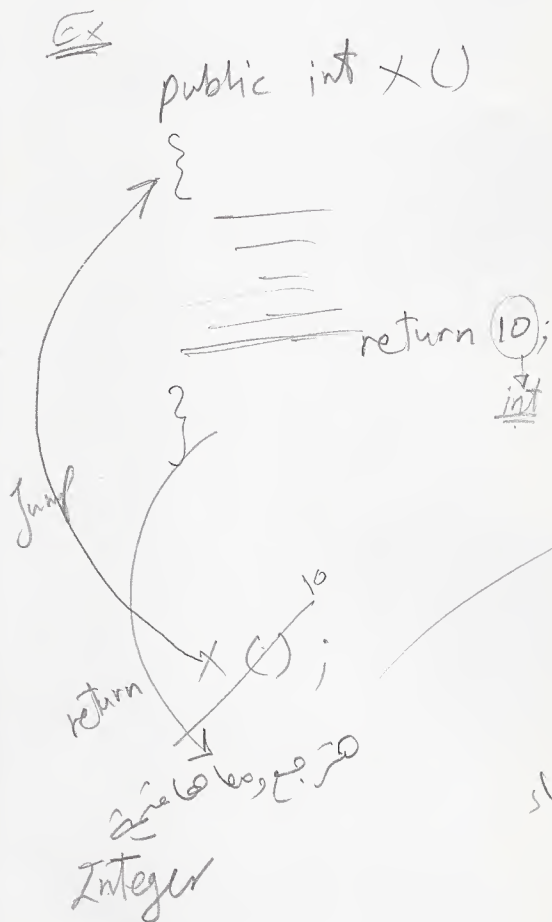
```

{
}

```

معنى النوع عندما يأتي قبل اسم الدالة

ج/ الدالة لا ترجع وقت الـ return هتارجع ومساها عدد integer



الدالة x لما تيج اشتدكلي
هذه الشكل:

x () ;

معنى return : روح ارجع مكانه الـ اشتدك وارجع الـ اشتدك
وعوضه بالشيء الـ

Lesson 25

الصفة العامة

```
public type name()
{
}
```

return

```
public int speed()
{
    int a = 10;
    int b = 20;
    int c = a + b;
    return c;
}
```

return معناها - ارجع قيمة الاسم الذي هو موجود مكانه بالقيمة

Lesson 26

```
public void welcome()
{
}
```

نوع الدالة void

* لا يتم وضع دوال النوع void في عملية الحساب

void =

الدالة لن ترجع قيمته وقت الرجوع
فهي كما هي الدالة مثل هل ترجع قيمة
عنه كذا لا يستخدما فيها return

Lesson 27

مفهوم ال parameters

argument

parameters: عبارة عن متغيرات يتم وضعها بين أقواس الدالة ويتم التحويل عن قيم هذه المتغيرات أثناء استدعاء الدالة

Sum(10,12) هنا متغيرات تظهر للدالة كما هي في التحويل عن قيمها أثناء الاستدعاء

ملحوظة: أي مجهول يحتاجه ال action يمثل ال parameter

Lesson 28

```
public class Car {
    public int model;
    public void speed(int s)
    {
        system.out.print ("speed is "+s);
    }
}
```

مثال مع Method

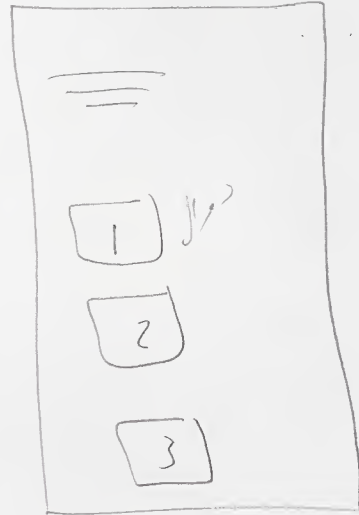
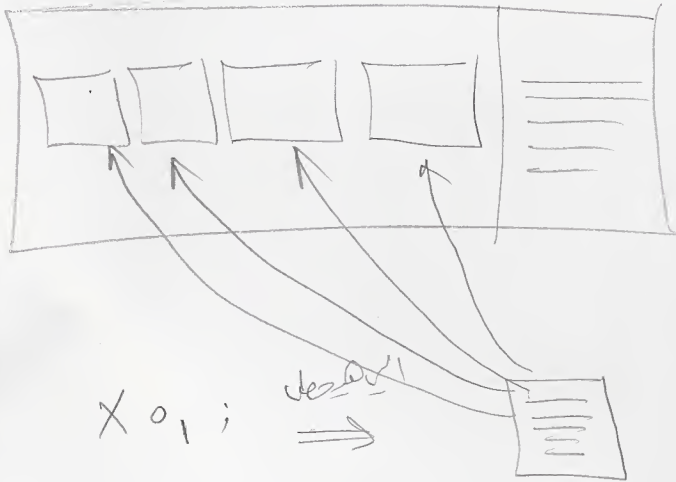
Lesson 29

معلومات أكثر عن ال class

مم

O₁ O₁

Class X



معلومات المتغيرات

والسؤال هيسم المتغيرات من ال class الاسمي
بديل ما ننسخه

الا شياء مشتركة تظل مكانها والفرق مشترك يتم نسخها

Lesson 30

مع معلومات تتركب بين المتغيرات والموال

action: $\xrightarrow{\text{تغير}}$ قيم المتغيرات



بعض الموال تقدم بتغيير قيم المتغيرات
او ارجاعها

Lesson 31

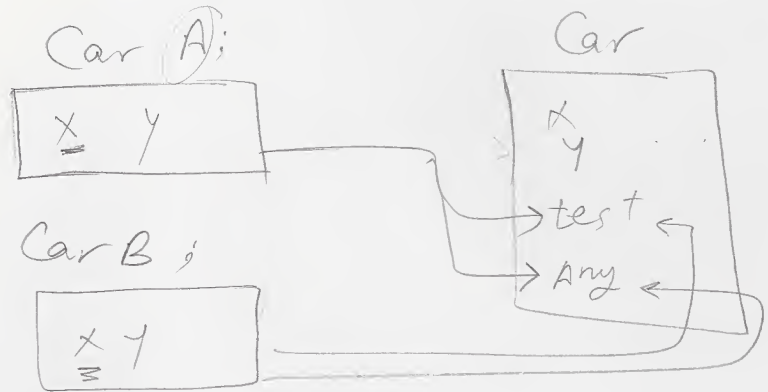
this المتغير

this: تستخدم للوصول لمحتويات الـ Class داخليا .
this: لا تستخدم خارج الكلاس .

Lesson 32

this

برمجة
على نسخ: instance



⊗ عناية بفرق بين x الموجودة في A و x الموجودة في B

⚠ this:

متغير يتم استدعاه وقت انشاء الكائن
باسم الكائن

+ this: متغير مجهول سيحل الكائن الذي يتم انشاءه

Ex:

```
public class Car {
    public int x;
    public int y;
    public void test()
    {
```

`this.x = 10;`

`this.y = 500;`

`}`

```
public void any()
{
```

`this.test();`

`}`

`}`

In case of Car A:

`A.x = 10;`

`A.y = 500;`

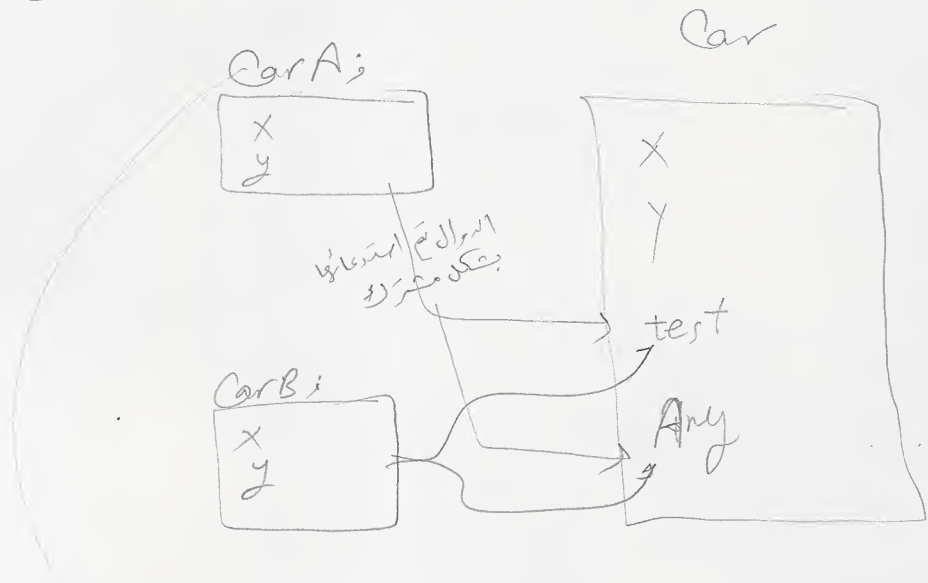
In case of Car B:

`B.x = 10;`

`B.y = 500;`

Lesson 33

المراد مع this



س/ كيف نعرف ان test تستخدم مع الكائن A أو الكائن B ؟
ج/ باستخدام متغير this

this.test(); تستخدم → A.test
معناها: طبق الدالة test مع متغير الكائن A

Lesson 34

مثال

Lesson 35

لإنشاء كائن : new

أول اسم الكلاس

Car Camry = new Car();

أكتب اسم الكلاس وكائن الدالة

⇒ object Camry from the class "Car"

new Car :- إنشاء نسخة جديدة من الكلاس
Car وضربها في الكائن Camry

~~طريقة أخرى لكثافة كائن في~~
~~Car Camry~~
~~خطأ~~

* دالة ال Constructor *

```
public Car ()
{
}
}
```

Constructor *

* اسم دالة ال Constructor عادة تعرفها "تتميزها" ← هيكون اسمها بنفس اسم الكلاس

* لا يوجد نوع إرجاع void لأنها تلقائياً ترجع object .

* وظيفتها :-

* تستدعي وقت إنشاء الكائن .

⚡ ملحوظة :- إذا لم يتم كتابة دالة ال Constructor في الكلاس ← سيتم إنشاء Constructor افتراضياً بواسطة Java

* يستخدم لتوفير القيم : حين أول ما يتم إنشاء الكائن ← أثناء لحظة الإنشاء

ماذا تريد أن تفعل ؟!

مثلاً : $this.speed = 0$; حينها سرعة = 0
استدعاء ال object .

Constructor & parameters

Car.java :-

```
public Car (int s)
{
    this.speed = s ;
}
```

Java Course . Java

Car Camry = new Car (60) ;

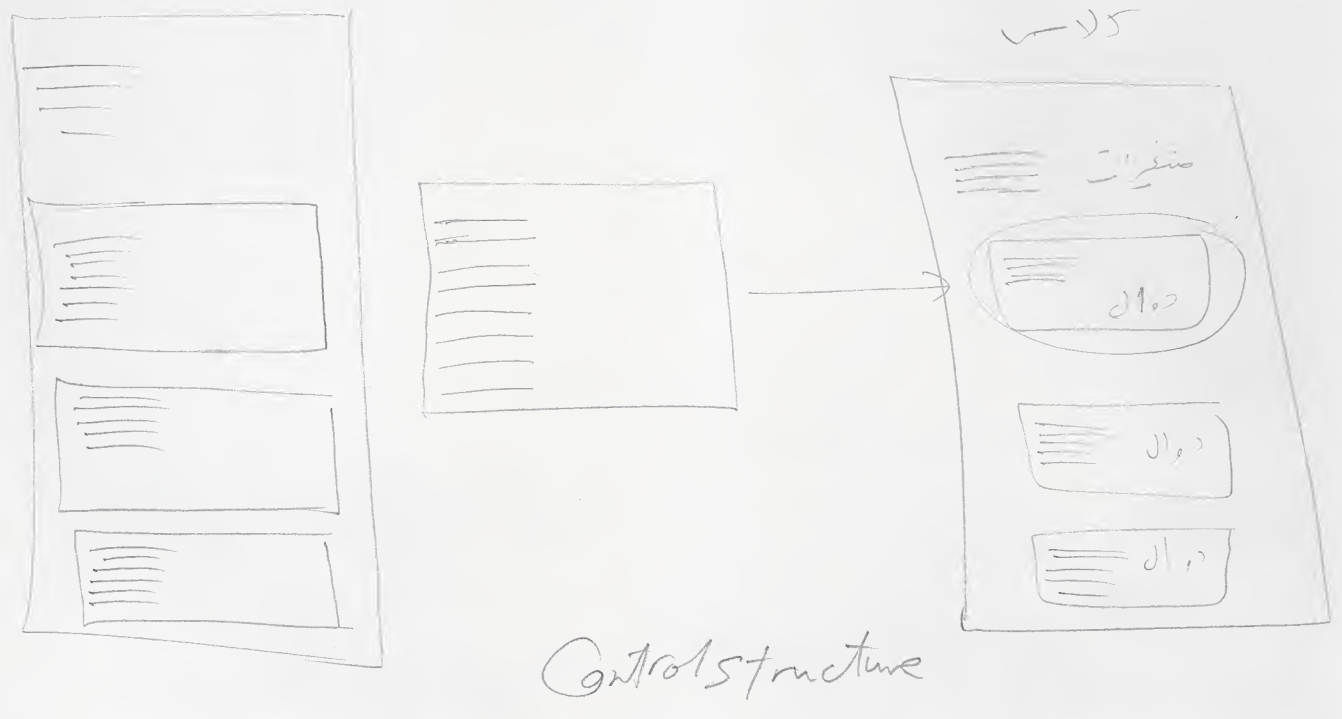
* دالة ال Constructor بها إنا دالة
يمكن أن تأخذ parameters

Lesson 38 :-

مکان بیضا مع کلاست
والطباعت
print

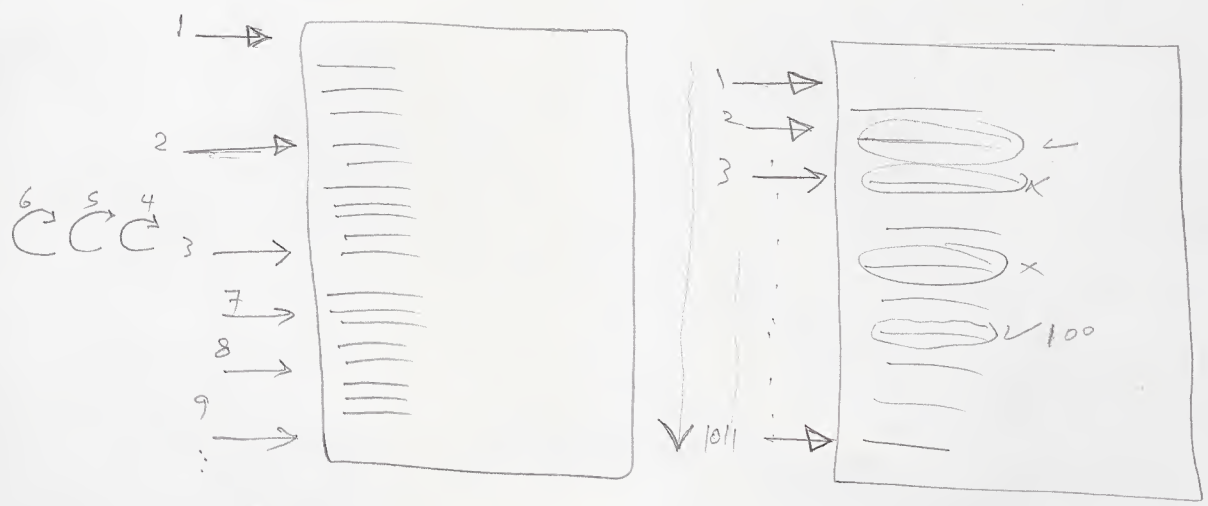
Lesson 39 :-

مخفف حول الكائنات والكلاسات



Lesson 40

مفهوم التحكم
Control Structure



التحكم في سير التنفيذ "أوامر التنفيذ"
Control Structure
أوامر التنفيذ وأوامر التنفيذ
أوامر التكرار وأوامر التكرار...

Lesson 41:-

استخدام If Condition

Control structure ←

If (شرط)

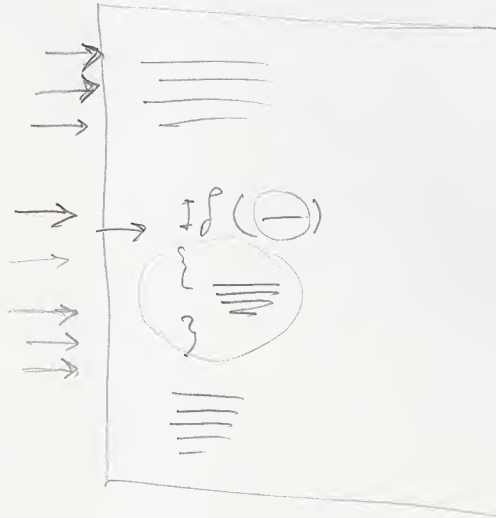
{

=====

}

true

لو الشرط صحيح سينفذ ما بين الأقواس
و لو لم يتحقق الشرط سينفذ ما بعد If
عجل قفز لبدء If كلها.



Ex: `int a = 5;`

If (a > 10)

{

=====

}

كره كغيرك كل الكود ده
لانه لم يتحقق الكود

Lesson 42:-

مثال مع استخدام If

Lesson 43

استخدام الالة else

if (شرط)

{
}
}



في حالة تحقق الشرط



else

{
}
}



في حالة عدم تحقق الشرط
نقدر الكود ده

Lesson 44

حالات else

Lesson 45

استخدام والمقارنات في جافا

أكثر 100 >

أقل <

يساوي ==

لا يساوي !=

أكبر من أو يساوي >=

أصغر من أو يساوي <=

دي المعادلة الرقيقة

< 200 || > 100

&& < 200 & > 100

! < 200

لا يساوي نفسها

b2.equals(b3)

!b2.equals(b3)

المساواة حرفية

لها درجتها بعد ذلك

يعني أو

يعني و

Lesson 46

else if أو إذا كانت

if (—)

{
}
}

else if (—)

{
}
}

else if

كذا if

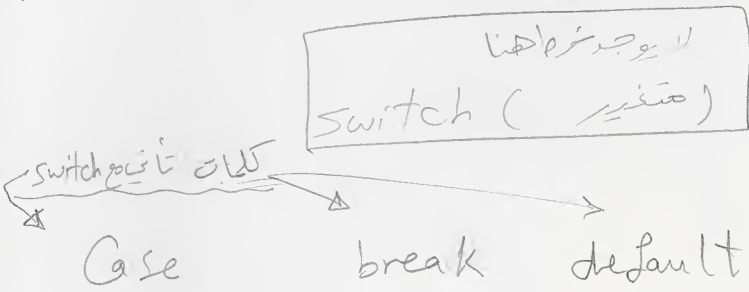
else

{
}

ويمكن بدون else

Lesson 47

switch



ملاحظة: لازم نضع Break; للفروع والا سيستم بتنفيذ الكود الآخر

قيمة
اختبر المتغير الموجود بين القوسين
هل فيه قيمة X

switch (x)
{

Case 1:

==
break;

لو نفذ الكود هذا
مخرج النتائج

Case 70:

==
break;

لم يبقها
هل هي 70؟
اذا 70 نفذ الكود

}

Lesson 48

استخدام switch مع أكثر من Case

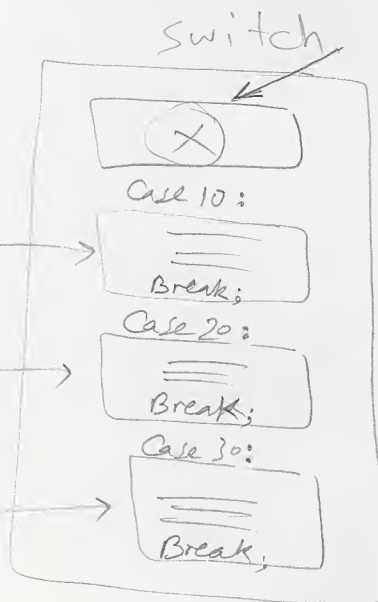
بيكليه على قيمة x

لو كانت 10 نفذ الكود ده

لو كانت 20 نفذ الكود ده

لو كانت 30 نفذ الكود ده

عني سيتم تنفيذ بلوك ما هيفقوا
وليس كلها



دي الحالة المباشرة

switch(x)

```
{  
  Case 10 :  
  Case 20 : } →
```

إذا كان أ أو ب
ننجز البوك هذا

```
break;
```

Case 30 :

```
break;
```

```
}
```

Case : تبدأ البوك

break : تنهي البوك

Lesson 49

مثال

Lesson 50

استخدام Default

مثال

حين إذا لم يجد ولا مورد توافق المطالب يتنجز الـ default

Lesson 51

استخدام أكثر من Case

Lesson 52

استخدام Inline-If

عبارة عن If ثنائية، ولكن مختصرة.

```
if (s=10)
    system.out.println("up");
else
    system.out.println("Down");
```

int c=10; else

int y = (c==10) ? 5 : 70

الشروط

لو لم يتحقق الشرط

لو تحقق الشرط

لو c=10 يبقى الشرط تحقق إذا y=5

لو c≠10 لم يتحقق الشرط إذا y=70

ويمكن شرط بأخذ شرط

Lesson 53

مفهوم loop / الحلقة / التكرار

وظيفة: تكرار أمر معين عدد محدد من المرات.

أو تكرار مجموعة من الأوامر عدد معين من المرات.



Lesson 54 :- & Lesson 55

عداد
Initialization
الشروط
Increment, decrement part
زيادة والنقصان

`for (int i = 1 ; i < 10 ; i++)`
استمر في اللولب طالما أن $i < 10$
زيادة بواحد
or $i = i + 1$

{

}

Lesson 56

مفهوم الزيادة في
For

`for (int i = 10 ; i >= 1 ; i--)`
System.....

{

}

Lesson 57

مفهوم الـ while
For

آلية عمل

{

Lesson 58

infinite loop
لولب غير منتهية

`int i = 0 ;`
`for (; ;)`
لازم تحديد
مخرج الخروج
→ $i \leq 10$

{

else
break ;
كسر اللولب

لازم تحديد
Infinite
→ $i++ ;$

Lesson 59 :-

معلومات عامة عن For

Lesson 60 :-

مكان مكامل عن For
يستخدم loop داخل loop

Lesson 61 :-

طباعة loop بشكل متتالي

Lesson 62 :-

طالما الشرط متحقق
يتم تنفيذ الكود
int x=0
while()

while loop

* يعني loop انما شرط فقط
والمعلومات الأخرى مسؤلية ليدرك أنت
و يتم تنفيذ الكود طالما أن الشرط متحقق

← هنا تخرج
ولا تكون Infinit
i++

← يعني لازم تكتب i++ هنا تخرج

Lesson 63 :-

مكان + معلومات إضافية

هام :-

ال while زي ال if بالضبط
ولكن ال if لن يرجع مرة أخرى لفوق
بينما ال while هيرجع لفوق بعد ما الشرط لا يتحقق

Lesson 64

do while

int i = 1;

do
{
system.out.println(i)
}
i++
while(i < 4);

* اكسها while مقولبة
+ while بتكتب بعد الأقواس في من البداية ومعلها فاهلة
منقولة while;

الفرق بينها وبين ال while
أن ال do while كسندف الكود مرة واحدة مع الأقل
مشاه الشرط بيكون بعد الكود

Lesson 65

Java loop

خلاصة

for (; ;) { }

إذا كانت النهاية معلومة

while () { }

غير معلومة

do { } while ();

نزي ال while وإذا استخدم
الكود مرة واحدة مع الأقل

Lesson 66

أنواع البيانات

char z = 'z';

لازم يتم وضعه بين تنقيط مفرد

حرف واحد فقط
لا حرفين

Lesson 67

String

String name = "Hi!";

System.out.println(name);

جافا كلاس خاص بال String مسبق مع الفيز
وبالتالي له دوال ومميزات يمكن استخدامها

Lesson 68

تابع دالة charAt()

Cx String name = "Anan";

Char c = name.charAt(1);

System.out.println(c);

كله هيطبع حرف n

ملاحظة هامة :-

في حالة ال String

يبدأ العد من الصفر

وظيفة ال charAt() طباعة حرف معين في حسب رقمه

Lesson 69

String تابع

length

طريقة طول النص

name.length();

الناجئ int القيمة

Lesson 70

طريقة length() مع CharAt()

Lesson 71

الناجئ، new String

String value = new String("anan");

Lesson 72

طريقة uppercase مع String

على كائنة تحول النص حروفه كابتال

name.toUpperCase();

Lesson 73

toLowerCase();

Lesson 74

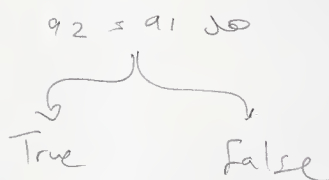
دالة مساواة النص حرفياً

String a1 = "aaa";

String a2 = "aaa";

a1.equals(a2);

طريقة string مع string



Lesson 75

equal & ignore case

عني كره مش مهم الحروف كالمبتال أو small

عني "AAa" = "aaA" نادى

Lesson 76

Array المصفوفات

نوع $\text{int } x = 10$ كره $\rightarrow \text{X} \rightarrow \text{V}$

مصفوفة $\text{int } Z[]$ \rightarrow 

المصفوفة :- عبارة عن متغير اكبر من قيمته "يتوي مع اكثر من قيمة"
فيكون هو Index عني كل قيمة تأخذ رقم بداية من 0

* الوصول للقيمة في المصفوفة :-

ده كره متغير

$Z[0]$ \rightarrow نوع المصفوفة Z
وهي القيمة التي في مكانها

لا تكسب مربعات خاوية عني كره انك بتعرف مصفوفة

$Z[]$

Lesson 77

طريقة كتابة المصفوفات

`int z[] = new int[10];`

١- جزي ١٠ خانة
للمصفوفة \Rightarrow

منزلة المصفوفة
شعوري على ١٠ قيم

Lesson 78

ادخال القيم للمصفوفة

`int z[] = new int[4];`

`z[0] = 1;`

`z[1] = 50;`

`z[2] = 300;`

`z[3] = 1000;`

index
فهرس

تم الوقوف عند \Rightarrow لأن الكود بدأ من فهرس
يحيى يوجد ٤ أماكن من أول فهرس

Lesson 79

«تعبير لمصفوفة»

طريقة مختلفة لكتابة المصفوفات وادخال القيم

`int z[] = { 1, 50, 300, 1000 };`

ولكن يجب أن تكون القيم كـ `Integer` أو `double` على حسب البرنامج
دمك string أيضاً ← عادي \Rightarrow

Lesson 80

حل المسألة الأولى

المتغيرات r و z مصفوفات

int [] r;

int [] r, z;

int [] z;

==

المتغيرات

int r, z[]

int متغير نوعه

int مصفوفة نوعه

Lesson 81

int كبريد نوع عام للمصفوفة

int values[] = {1, 2, 3} مصفوفة مكونة من نوع المصفوفة

double y[] = {1.5, 1.3}

طول المصفوفة

Lesson 82

نمط مطابقة المصفوفة كما نرى كانت بداية دوال

values.length طريقة طول المصفوفة

Hint For (int i = 0; ; -)

Index
لأنه المصفوفة تبدأ من الصفر
من الصفر
المصفوفة
المصفوفة

Method
• value
• value

Lesson 83

مقالتي طول المصفوفة طول الدوال

String name = "anan";

int z[] = {1, 2, 3, 4}

System.out.println(name.length());

الطريقة المستخدمة في الدوال

System.out.println(z.length);

خاتمة

فيديو يوجد في القوائم


```
String values[] = { "anan", "Sara",  
                    "Mido",  
                    "anan"  
                    };
```

الترتيب ده حتميا فقط
شكل عين المبرمجين

```
System.out.println(values[0]);
```

← كده طبع anan

* كل قيمة في صفوفة حاد اكانت نصية او عددية تعتبر كأنها متغير واحد
عني values[0] يعتبر متغير يشير الى القيمة "anan"
عني اسم الصفوفة مع رقم الاندكس يمثل اسم متغير طالع مثلا
String name = "anan";

main

```
int values[] = new int[10]; // fixed size array
```

2:40
main

```
public int createArray(int size) {  
    int c[] = new int[size]; // variable length array  
    return c;  
}
```

نوع الارجاع: صفوفة نوعها:

لو لم يتم وضع [] بجوار int سيظهر خطأ لان صفوفة وارجع int وارجع
كود المتروك وارجع بين [] int لانه عام

Lesson 86

استخدام for each للمصفوفات

* تعتبر for each لوحة خاصة للتعامل مع المصفوفات

int values[] = {1, 4, 9};

for (int val : value) // for each

المتغير
اسم المصفوفة
بمكون أقواس

```
{
    System.out.println(x);
}
```

كده هيطبع
1
4
9

* نفس for العادية لكنه يعد لوحة
فيعرض جميع عناصر المصفوفة
كده هيعر لوحة

* يعني : كل مرة تأخذ لغة واحدة
من المصفوفة values يعني هذا العنصر
1 وخرزته في val

ثم في اللغة الثانية هذا العنصر 4 وخرزته في val

* يعني يا فتها for each فكرتها انك بتعرف متغير
وهي تعرف بانك الـ value مصفوفة فبالثاني تعرف طولها
فأنت لا تحتاج تعرف طولها كرام

(الآن لغة بتخزين value القيمة في val)

Lesson 87

تصدير مصفوفات الدوال

مصفوفة اسمها aa
x.printarray(aa)
وليست
x[aa]

كبارا مبرمجة
اسم المصفوفة يُمرر للمالة بعبارة أقواس []
انظر الفيديو للفهم

Lesson 88

هل for each تُغير القيم؟

* for each لا تغير من قيم المصفوفة ولكنها تأخذ نسخة وقد عمل عليها براهين
ولكن القيم الأصلية لا يتغير التعديل عليها

+ عمل مصفوفة من الكائنات طبيعي مثل عمل أي نوع آخر

~~Car~~ Car y[] = new Car[10];

+ أنشئ مصفوفة

اسم y[] من الكلاس Car
وهو مكتوب 1. كائنات من Car

y[0].setSpeed(100);

y[1].setSpeed(200);

⋮

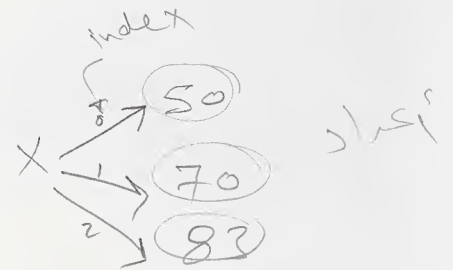
لكن كده غير صحيح
فيجب تعريف عناصر مصفوفة بإنشاء

y[0] = new Car();

y[1] = new Car();

Multi-dimensional array "2D"

Ex: int[] x = new int[3] ⇒

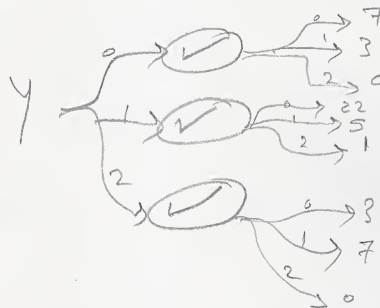


x[0] ⇒ 50 ✓



+ يعني الفتح نفسه عبارة عن مصفوفة وليس عدد لأنني 2D

عني يعني كذا كذا



أنشأ مصفوفة نوعي int

وكان عنصر فيها عبارة عن مصفوفة
بنوع int

Ex: $\text{int} \llbracket \llbracket \llbracket \rrbracket \rrbracket \gamma = \{ \{1,2\} \{3,4\} \}$

تقسيم المصفوفة الأولى

التعريف 2D :-

* المصفوفة الكبيرة عبارة عن عنصرين وكل عنصر عبارة عن مصفوفة أخرى
 * كل عبارة عن مصفوفة كل عنصر فيها عبارة عن مصفوفة أخرى.

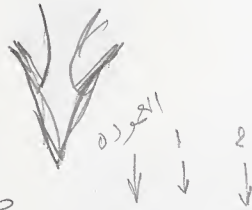
توضيح على المصفوفات 2D

Lesson 91 :-

$\text{int } m \llbracket \llbracket \llbracket \rrbracket \rrbracket = \{ \{1,5,6\}, \{5,7,9\}, \{3,1,5\} \}$

صفوف
rows

عمود
col.



$\text{int } m \llbracket \llbracket \llbracket \rrbracket \rrbracket = \{ \begin{matrix} \text{المصفوفة 0} \\ \rightarrow \{1,5,6\} \end{matrix}, // \text{المصفوفة 1} \\ \rightarrow \{5,7,9\}, // \text{المصفوفة 2} \\ \rightarrow \{3,1,5\} \}$

إشارة مصفوفة متغيرة باستخدام new

Lesson 92

Ex

مصفوفتين يمكن مصفوفة بها 2 عناصر $\text{int } \gamma \llbracket \llbracket \llbracket \rrbracket \rrbracket = \text{new int } \llbracket 2 \rrbracket \llbracket 4 \rrbracket$;

$\gamma[0][0] = 3$;
 $\gamma[0][1] = 5$;
 $\gamma[0][2] = 13$;
 $\gamma[0][3] = 15$;

$\gamma[1][0] = 9$;
 $\gamma[1][1] = 8$;
 $\gamma[1][2] = 7$;
 $\gamma[1][3] = 4$;

Lesson 93

اللووب المصفوفة المتعددة

* المبرمجين يستخدمون اثنين لوب للعامل مع المصفوفة الثنائية

$\text{int } x[][] = \left\{ \begin{array}{l} \rightarrow \{11, 53, 66\}, \\ \rightarrow \{9, 71, 63\}, \\ \rightarrow \{7, 68, 69\} \end{array} \right\};$
تصايف "مصفوفات"

for (int i = 0 ; i < x.length ; i++)

{ int z[] = x[i];

for (int y = 0 ; y < z.length ; y++)

{ system.out.println(z[y]);

}

}

كده صليبع كل عناصر المصفوفة الثنائية

Lesson 94

مثال للووب المصفوفة المتعددة
مثال مبسط عن السابق

كلام جازي للفرم

Lesson 95

استخدام Float و Double

float مصفوفة

* هذه من ضمن أنواع البيانات Data types

float x = 1.2f

عدد بفاصلة عشرية
"4 byte"

Double y = 1.2543211112

عدد بفاصلة عشرية أدق
و لكن حجمه أكبر "8 byte"
في الذاكرة من ال float

يتم كتابة f في حالة float لأن الاختلاف بين الأرقام العشرية
هو ال double لذلك يتم إخبار الكمبيوتر بأن عدد float يرفع
حرف f بعد الرقم .

Lesson 96

استخدام short و long

byte
short $\xrightarrow{\text{يا هذا}} 16 \text{ bit} = 2 \text{ byte}$
int $\longrightarrow 32 \text{ bit} = 4 \text{ byte}$
long $\longrightarrow 64 \text{ bit} = 8 \text{ byte}$

كل واحد
أعداد
بمئة

القيم ما بين السالب والموجب

Lesson 97

استخدام النوع Boolean

Boolean x ;

عبارة عن متغير يأخذ قيمتين :-

true or false
صح خطأ
1 في الكمبيوتر
0

غير ينفع في الشيء الذي به حالتي

"أما أو"

+ ينفع في الشرط مثلا / ذكر - أنثى / حي - ميت .

Lesson 98

استخدام النوع Byte

byte $\xrightarrow{\text{يا هذا}} \text{عدد صحيح} \xrightarrow{\text{من}} -128 \text{ إلى } 127$

ومعلوم بأن البايته في الكمبيوتر 256 خانة تبدأ من صفر وتنتهي عند 255

+ دائما يستخدم في البرمجيات التي تتعامل مع الحارديز

+ يستخدم في التشفير وصناعة الفيديوهات

+ وفي البرمجيات التي تتعامل مع الذاكرة

Lesson 99:-

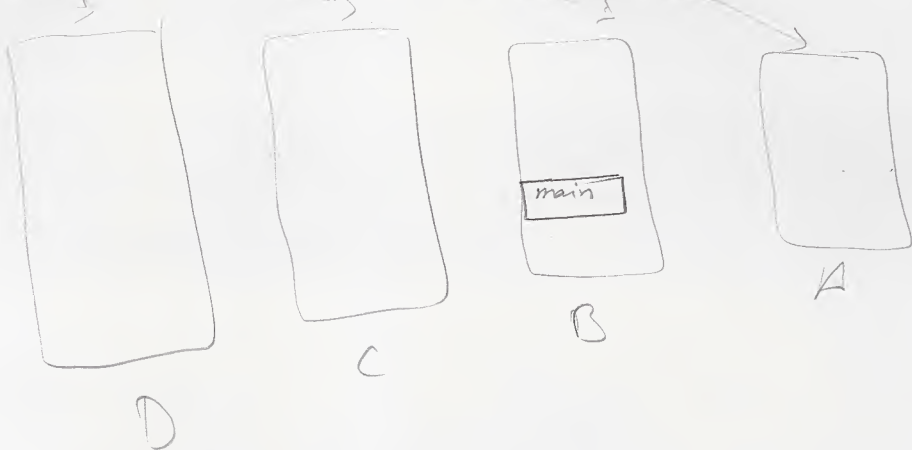
توضيح مهم



Lesson 100:-



معلومات عن أول main



الدالة لا يمكن أن تكون:

1. غير تعريف

2. استدعاء

* عند الفتح Run ربياً تنقل البرنامج

تقوم جافا بالبحث عن دالة `main`

وأول ما جافا تلمس `main` سيد التنفيذ فوراً فتعدها

* تسمى دالة `main` ← `entry point` نقطة البدء/نقطة الدخول

* جافا أول ما تلمسها تذهب إليها `main` وأخيراً تخرجها من `main`

Lesson 101:-

مثال على الجمع

* شاهد الفيديو

* بإمكانه إشار كائن من كلاس بنفس الكلاس «يعني داخله»

Lesson 102

مثال على إشار كلاس Calc

* عند إنشاء مشروع جافا أنت بتعمل مجموعة ملفات جافا وتستدعيها من طرف

تعريفها بدائل بعضها

* طريقة التواكل بين ملفات المشروع ← عن طريق تعريف كائنات بدائل الكلاسات بعضها لبعض

مثال على خاصية دالة

Lesson 103 :-

مثال على التعامل مع التكرار

site.length() - 1

عنايه الكريبير من حرف

Lesson 105 :-

مثال لتحويل بعض اى معلومات

Lesson 106 :-

خرب عنام معلومات

* لانهم تكون معلومات في النوع هي ٢x٢ أو ٣x٣ " قبل الطول "

Lesson 107 :-

عرض مشكلة خرب المعلومات

Lesson 108 :-

مثال لكلاس اكتاب

Lesson 109 :-

* برمجة الكائنات تعتمد على الوهم الدقيق .

* بالانكليزية تعريف اى كائن دافد اى كلاس

وكذلك استخدام دالة وصفية .

Lesson 110 :-

Lesson 111 :-

* هذا الجزء من الدورة كانه مدرج كيفية بناء

Lesson 112 :-

المجلد وقراءتها Syntax

Lesson 113 :-

Lesson 114 :-

Lesson 115

Lesson 116

Lesson 117

Done